IN THE CLAIMS:

1. (currently amended) A method for executing an interrupt in a data processing system comprising the steps of:

fetching a conditional store instruction that is conditional upon a reservation;

receiving notice that an interrupt is pending in the data processing system, wherein the receiving the notice that an interrupt is pending further comprises starting a counter in response to receiving the notice;

invalidating the reservation in response to receiving the notice, wherein invalidating the reservation causes the conditional store instruction to finish; and

processing the interrupt.

2. (canceled)

3. (currently amended) The method of claim ~~2~~ 1, further comprising the steps of:

determining that the conditional store instruction is non-speculative and the oldest entry of a store queue; and

reporting completion of the conditional store instruction before the counter reaches a predetermined count value.

4. (original) The method of claim 3, further comprising the step of flushing a completion unit queue and beginning the processing of the interrupt.

5. (currently amended) The method of claim ~~2~~ 1, further comprising the steps of:

determining that the conditional store instruction is not the oldest entry of a store queue;

converting the conditional store instruction to a nop instruction; and

removing the conditional store instruction from a completion unit queue when the counter reaches a predetermined count value.

6. (original) The method of claim 5, further comprising the step of flushing a completion unit queue and beginning the processing of the interrupt.

7. (original) The method of claim 1, wherein the data processing system has one or more processors.

8. (original) The method of claim 1, wherein the conditional store instruction is an instruction that requires a corresponding reservation of a memory location, wherein the corresponding reservation was established by a previously executed load and reserve instruction.

9. (original) The method of claim 1, wherein the step of setting a reservation related to the conditional store instruction comprises setting an address and a valid bit in a reservation register corresponding to a location in a memory for the conditional store instruction.

10. (currently amended) A data processing system, comprising:

a processor for executing instructions, the processor comprising:

a memory unit;

an instruction dispatch unit for fetching, decoding, and issuing a conditional store instruction; ~~and~~

a reservation register for storing a reservation corresponding to a location in the memory unit to be used as a target for the conditional store instruction, wherein in response to the data processing system receiving an interrupt, the reservation is cancelled; and

a completion unit having a counter and an instruction queue, wherein the counter is started in response to receiving the interrupt, and wherein the instruction queue is flushed when the counter reaches a predetermined count value allowing processing of the interrupt to begin.

11. (canceled)

12. (currently amended) The data processing system of claim ~~11~~ 10, further comprising:

a store queue having an oldest entry, the store queue for temporarily storing the conditional store instruction until the conditional store instruction is no longer speculative and any store instructions ahead of the conditional store instruction are performed, wherein when the conditional store instruction is the oldest entry of the store queue when the interrupt is pending, completion of the conditional store instruction is reported before the counter reaches a predetermined count value.

13. (original) The data processing system of claim 12, further comprising a condition code register for storing a state of the processor, wherein the state of the processor is updated with the success/failure of the conditional store instruction.

14. (currently amended) The data processing system of claim ~~11~~ 10, further comprising:

a store queue having an oldest entry, the store queue for temporarily storing the conditional store instruction until the conditional store instruction is no longer speculative and any store instructions ahead of the conditional store instruction are performed, wherein when the conditional store instruction is not the oldest entry of the store queue when the interrupt is pending, the conditional store instruction is converted to a nop instruction.

15. (currently amended) The data processing system of claim ~~11~~ 10, wherein the data processing system further comprises:

a second processor for executing instructions, the second processor comprising:

a second memory unit;

a second instruction dispatch unit for fetching, decoding, and issuing a second conditional store instruction; and

a second reservation register for storing a second reservation corresponding to a location in the second memory unit to be used as a target for the second conditional store instruction, wherein in response to the second processor receiving a second interrupt, the second reservation is cancelled.

16. (currently amended) A data processing system comprising:

a system bus;

a memory coupled to the system bus;

a first processor, coupled to the system bus, for executing instructions, the first processor comprising:

a first instruction dispatch unit for fetching, decoding, and issuing a first conditional store instruction; ~~and~~

a first reservation register for storing a first reservation corresponding to a location in the memory to be used as a target for the first conditional store instruction, wherein in response to the first processor receiving an interrupt, the first reservation is cancelled; and

a completion unit having a counter and an instruction queue, wherein the counter is started in response to receiving the interrupt, and wherein the instruction queue is flushed when the counter reaches a predetermined count value allowing processing of the interrupt to begin; and

a second processor, coupled to the system bus, for executing instructions.

17. (canceled)

18. (currently amended) The data processing system of claim ~~17~~ 16, wherein the first processor further comprising:

a store queue having an oldest entry, the store queue for temporarily storing the first conditional store instruction until the first conditional store instruction is not speculative and any store instructions ahead of the first conditional store instruction are performed, wherein when the first conditional store instruction is the oldest entry of the store queue and the interrupt is pending, completion of the first conditional store instruction is reported before the counter reaches a predetermined count value.

19. (currently amended) The data processing system of claim ~~17~~ 16, further comprising:

a store queue having an oldest entry, the store queue for temporarily storing the conditional store instruction until the conditional store instruction is no longer speculative and any store instructions ahead of the conditional store instruction are performed, wherein when the conditional store instruction is not the oldest entry of the store queue and the interrupt is pending, the conditional store instruction is converted to a nop instruction.

20. (original) The data processing system of claim 16, wherein the second processor further comprising:

a second instruction dispatch unit for fetching, decoding, and issuing a second conditional store instruction; and

a second reservation register for storing a second reservation corresponding to a location in the memory to be used as a target for the second conditional store instruction, wherein in response to the second processor receiving an interrupt, the second reservation is cancelled.